

In the Claims:

Please amend Claims 1-7 and 10-16; cancel Claims 8 and 17; and add new Claim 19, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented claims in a continuing or future application.

1. (Currently Amended) A system including an integrated development environment for use with a [[JMS]] mark-up language, comprising:

an integrated development environment that includes a graphical user interface that executes on a client machine, and that allows a user to enter a markup language program;

a parser that receives the markup language program from the integrated development environment and parses the markup language program to extract markup language commands;

a command processor that validates the markup language commands, and, for each markup language command converts the markup language command into a command object for communication to a command dispatcher;

a command dispatcher that receives command objects from the command processor and, for each command object, assigns the command object to one of a plurality of categories corresponding to a plurality of application program interfaces;

a plurality of processor modules, including a processor module for each category of application program interface, wherein each processor module receives the command objects assigned to its category, and performs appropriate operations against the corresponding application program interface; and

wherein the integrated development environment allows [[a]] the user to edit and modify the markup language programs program [[that]] as desired to access [[JMS]] the application program interfaces.

2. (Currently Amended) The system of claim 1 wherein the markup language is JMSML Java Message Service Markup Language.

3. (Currently Amended) The system of claim 1 wherein the graphical user interface includes a source editor that allows a user to enter programs as [[XML]] Extensible Markup Language code.

4. (Currently Amended) The system of claim 1 wherein the graphical user interface includes a design editor and a set of toolbars that allow a user to generate [[XML]] Extensible Markup Language source code by visually assembling [[JMS]] commands conforming to the Java Message Service specification.

5. (Currently Amended) The system of claim 2 wherein the graphical user interface includes a source editor that allows a user to enter JMSML Java Message Service Markup Language programs as [[XML]] Extensible Markup Language code.

6. (Currently Amended) The system of claim 2 wherein the graphical user interface includes a design editor and a set of toolbars that allow a user to generate JMSML Java Message Service Markup Language [[XML]] Extensible Markup Language source code by visually assembling JMSML Java Message Service Markup Language commands.

7. (Currently Amended) The system of claim 1 further comprising a parser that parses said program and communicates said markup language components to a command processor; and,

a command processor that converts the markup language components into operations conforming to one of [[JMS]] the Java Message Service or [[JMX]] the Java Management Extensions system operations specifications.

8. (Canceled).

9. (Original) The system of claim 1 wherein said integrated development environment is used to communicate said markup language components to said remote server via a wide area network or the Internet.

10. (Currently Amended) A method of using an integrated development environment with a [[JMS]] mark-up language, comprising:

providing an integrated development environment that includes a graphical user interface that executes on a client machine, and that allows a user to enter and edit a markup language program;

receiving the markup language program from the integrated development environment and parsing the markup language program to extract markup language commands;

validating the markup language commands, and, for each markup language command converting the markup language command into a command object for communication to a command dispatcher;

receiving command objects at the command dispatcher and, for each command object, assigning the command object to one of a plurality of categories corresponding to a plurality of application program interfaces;

communicating the command objects to a plurality of processor modules, including a processor module for each category of application program interface, wherein each processor module receives the command objects assigned to its category; and[.]

performing appropriate operations against the corresponding application program interface, as specified by the user in the accepting commands from a user to edit and modify markup language programs program that access JMS interfaces.

11. (Currently Amended) The method of claim 10 wherein the markup language is JMSML Java Message Service Markup Language.

12. (Currently Amended) The method of claim 10 wherein the graphical user interface includes a source editor that allows a user to enter programs as [[XML]] Extensible Markup Language code.

13. (Currently Amended) The method of claim 10 wherein the graphical user interface includes a design editor and a set of toolbars that allow a user to generate [[XML]] Extensible Markup Language source code by visually assembling [[JMS]] commands conforming to the Java Message Service specification.

14. (Currently Amended) The method of claim 11 wherein the graphical user interface includes a source editor that allows a user to enter JMSML Java Message Service Markup Language programs as [[XML]] Extensible Markup Language code.

15. (Currently Amended) The method of claim 11 wherein the graphical user interface includes a design editor and a set of toolbars that allow a user to generate JMSML Java Message Service Markup Language [[XML]] Extensible Markup Language source code by visually assembling JMSML Java Message Service Markup Language commands.

16. (Currently Amended) The method of claim 10 further comprising:  
parsing said program and communicating said markup language components to a command processor; and,  
converting the markup language components into operations conforming to one of [[JMS]] the Java Message Service or [[JMX]] the Java Management Extensions system operations specifications.

17. (Canceled).

18. (Original) The method of claim 10 wherein said integrated development environment is used to communicate said markup language components to said remote server via a wide area network or the Internet.

19. (New) A computer readable medium including instructions stored thereon, which when executed cause the computer to perform the steps of:

providing an integrated development environment that includes a graphical user interface that executes on a client machine, and that allows a user to enter and edit a markup language program;

receiving the markup language program from the integrated development environment and parsing the markup language program to extract markup language commands;

validating the markup language commands, and, for each markup language command converting the markup language command into a command object for communication to a command dispatcher;

receiving command objects at the command dispatcher and, for each command object, assigning the command object to one of a plurality of categories corresponding to a plurality of application program interfaces;

communicating the command objects to a plurality of processor modules, including a processor module for each category of application program interface, wherein each processor module receives the command objects assigned to its category; and

performing appropriate operations against the corresponding application program interface, as specified by the user in the markup language program.